| **비단길(Silk Road)** | |
| --- | --- |
| **13주차** | **2024.03.17 (일) ~ 2024.03.23 (토)** |

**■ 주간 회의**

**■ 다음 주 세부 계획**

**▶ [권순원]**

* 유니티 파티클을 d3d12에 알맞게 바꿀 수 있는 방법 찾기

**▶ [박준영]**

* 중간발표 스테이지 지형지물 제작 및 오브젝트들 박스콜라이더 수정
* ai 개발

**▶ [엄장헌]**

* 애니메이션 인스턴싱에 따른 문제 해결

**■ 작업 일지**

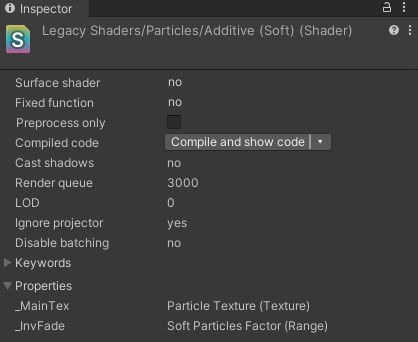
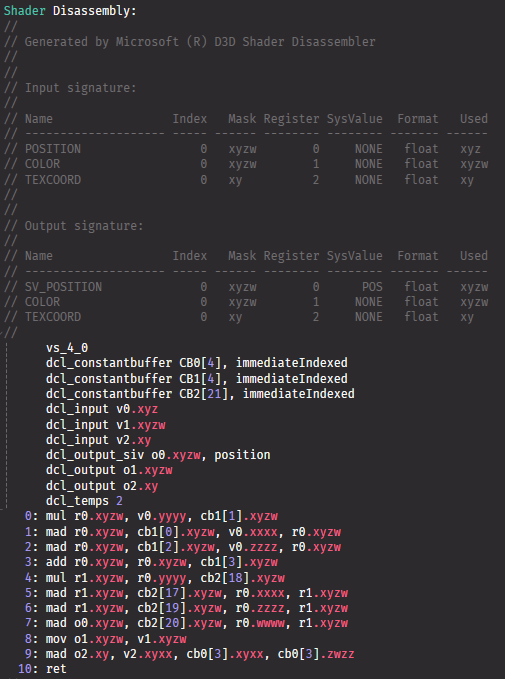
**▶ [권순원]**

일단 기존 파티클을 불러오는 것을 만들기 전에 프로젝트에 파티클 시스템을 제작해 보았다. 이전에 프로젝트에서 만들어본 파티클 효과를 참고해서 제작했다.

파일에서 읽어와 파티클을 생성하는 것이 아닌 일단은 텍스쳐만 파일로 읽어오고 코드 상에서 수치를 지정하는 방식으로 제작했다.

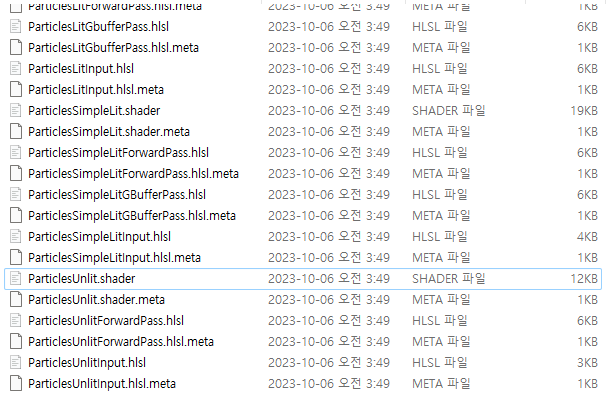
이후 유니티에서 파티클을 렌더링 하는 것에 대해 더 찾아보았다.

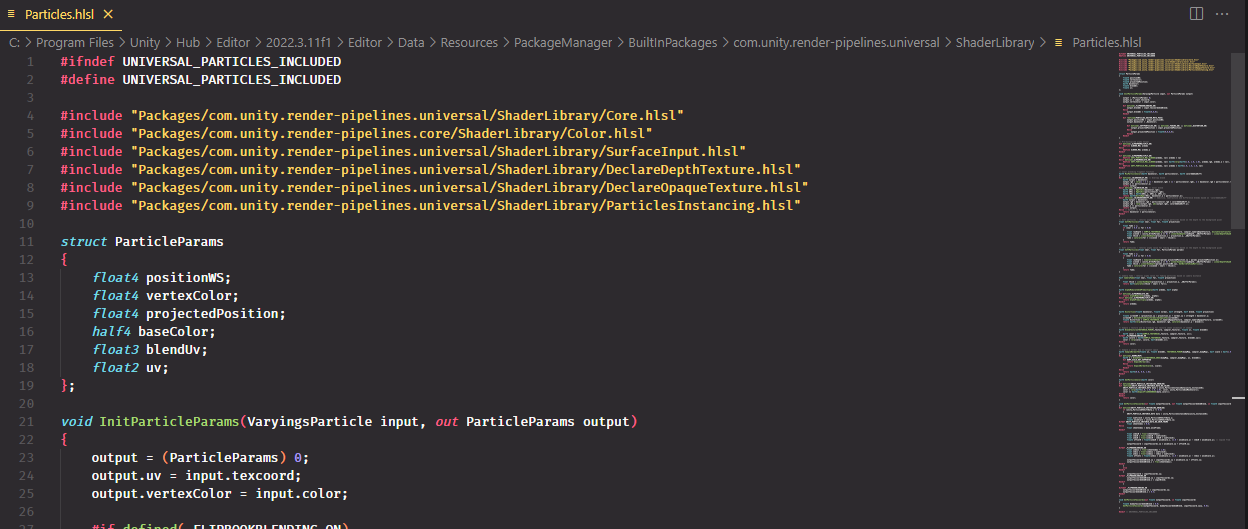
이전에 알아본 것처럼 유니티에서 파티클은 다른 오브젝트와 비슷하게 prefab으로 이루어져있다. xml 형식처럼 각 항목들이 있고 해당하는 값이 쭉 적혀있는 형식이었다. 그리고 본격적으로 파티클을 렌더링 하기 위한 셰이더도 확인해 보았다.

유니티는 d3d와 다르게 hlsl를 사용하지 않고 CG를 사용한다. 그래서 유니티에서 사용되는 셰이더 코드를 확인해보기로 했다.

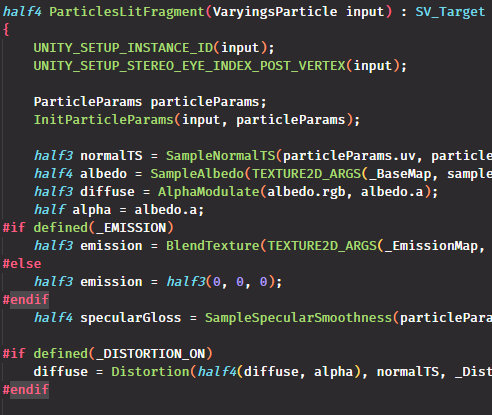
첫 번째로는 유니티 상에서 Compiled code를 볼 수 있는 기능이 있어 확인해봤다.

간단한 파티클이었는데 약 1000줄에 달하는 어셈블리 코드를 확인할 수 있었다. 이런 형태의 코드를 원한 것은 아니었기 때문에 직접 폴더를 검색해가며 찾아보았다.

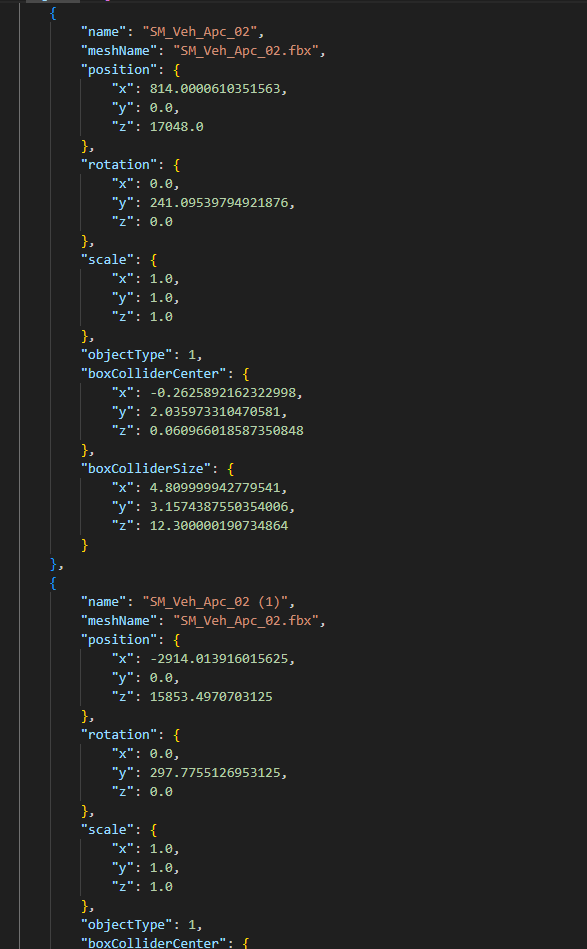




.shader형식의 파일과 생각과 다르게 .hlsl 형식의 파일을 확인할 수 있었고, Particle.hlsl 라는 파일을 찾을 수 있었다. 코드를 읽어봤을 때 내가 지금까지 보던 셰이더 파일과 달리 각 파이프라인의 함수가 없었다. 때문에 해당 파일은 파티클 셰이더를 만들 때 사용할 유틸함수들을 모아놓은 것이라고 생각했다.

다른 파일을 더 찾아 내가 알고 있는 셰이더 코드와 비슷한 파일을 확인했는데, openGL에서 사용해본 것과 비슷하게 Vertex와 Fragment로 이루어진 코드를 볼 수 있었다. 코드를 확인해보고 이 셰이더 코드를 d3d 셰이더와 호환되도록 자동으로 바꿔주는 스크립트를 만드는 것은 어려울 것 같다는 생각이 들었다. 

**▶ [박준영]**

****

맵에 오브젝트들을 배치한 후, 입맛에 맞게 박스콜라이더를 조정(우리 게임의 물리연산에 사용할 범위 수동지정)하여 이렇게 배치한 맵 오브젝트들의 모델링 정보, 포지션 정보, 타입, 콜라이더 정보 등등을 json으로 내보내는 유니티 툴을 개발하였다.. 유니티상에서 실제 게임에 적용될 맵을 모두 배치한 후, 원클릭으로 우리 게임에 그대로 적용할 수 있다.

json 직렬화 과정에서 오류가 발생했는데 유니티의 직렬화 기능에 미숙해있었다. 내보내려는 json파일의 구성요소들을 리스트화하여 해결하였다.

일부 계층적 구조의 fbx모델 로드에 문제가 있었으나 모든 계층의 오브젝트들을 불러와 그 개수에 맞게 조합시켜 맵상에 배치하는 기능을 구현하였다.

추가적으로 시험삼아 중간발표에 사용할 맵 오브젝트들을 배치해보았다.

이러던 중 일부 오브젝트가 텍스쳐를 살짝 이상하게 읽어들이는 현상을 발견하여 다같이 버그 수정작업중이다.

**▶ [엄장헌]**



준영이가 맵로드를 진행할때 콜라이더의 정보도 함께 가져오기로 했다. 아직 AABB만 만들어 회전했을때에 콜라이더도 함께 회전하지 않고 있었다. 따라서 OBB를 이용해 콜라이더를 만들었고 추가하였다.



다른 클래스들로 틀도 잡혀있고 단순하게 작업하면 될 것이라 생각했는데 쿼터니언 쪽에서 발목이 잡혔다. DXCollision에서 쿼터니언쪽에서 계속 오류가 생겼고 코드를 따라서 들어가보니 정상적인 쿼터니언이 아니라고 판단되었다. 쿼터니언에 대한 이해력이 부족해서 생긴 일이였고 degree에서 radian으로 만들고 XMQuaternionRotationRollPitchYawFromVector해당 함수로 변환을 해주어 정상적인 값이 적용되도록 수정하였다.

애니메이션 쪽 프레임 드랍 문제는 지금 며칠째 해결을 못 하고 있다.

어떤 방식으로 해결할지는 생각을 다 해두었는데, 코드만 바꾸면 계속 루트시그니쳐 관련해서 오류가 발생하고있다. 우선은 급하게 하기보단 루트 시그니쳐 쪽에서 천천히 다시 훑어봐야 할 것 같다.